BCCT (HONS) Cloud computing

BIE3262/BIE3264 • Major Project I & 2

Title : Queuing Numbering System Based Android

Submission Date : Week 16

Supervisor: MS. Suhaila

Student Name: Emad Jumaa

Student ID : 110045388

Term : Aug-Nov 2018

THE PROJECT REPORT IS PREPARED
FOR
FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY
IN PARTIAL
FULFILLMENT FOR

BCCT (HONS) CLOUD COMPUTING FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY

LIMKOKWING UNIVERSITY OF CREATIVE TECHNOLOGY

NOVEMBER 2018

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for undergraduate degree from Limkokwing University of Creative Technology, I agree that the University Library may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by my supervisor, in his absence, by the Dean of the faculty of Information & Communication Technology. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain should not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Limkokwing University of Creative Technology for any scholarly use which may be made of any material from my thesis.

Request for permission to copy or to make use of material in this thesis, in whole or in part should be addressed to:

Dean of the Faculty of Information & Communication Technology

Limkokwing University of Creative Technology

Jalan Teknokrat 1/1

63000 Cyberjaya

Selangor Darul Ehsan

Acknowledgment

I would like to thank my lecture, Ms. Suhaila, for the patient guidance, encouragement and advice she has provided throughout my time as her student. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly

Abstract

Queuing in visa department is an approach that lining up of students in visa hall in order to be served by Visa department employees at each counter. Students have to stay inside the visa department hall until there turn come without knowing time needed to be spent, some students even leave visa hall before, being served, This and other obstruction result too much delay in students waiting time the aim of this project is to minimize waiting time in queue by proper queue management and there by improve the high demand regarding on managing and serving student's needs. I developed an android application that will allow Students to join a virtual queue via mobile phone, and allowing them to check there turn without the need to be present at the visa hall also, providing an estimated time that gives the Students proper idea when their turn will start.

1 Table of Figure

Figure 1sample on related application.....10

Figure 2Use case diagram.....28

Figure 3Context diagram.....29

Figure 4Story Board.....30

Figure 5Flowchart diagram.....31

Figure 6Entity Relationship Diagram 32

Figure 7Dataflow Diagram.....33

Figure 8 LogIn Activity.....41

Figure 9 SignUp Activity.....42

Figure 10 Collect Number Activity.....43

Figure 11 LogIn Layout 44

Figure 12 SignUp Layout 45

Contents

ACKNOWLEDGMENT ..... **II**

ABSTRACT ..... **III**

1 TABLE OF FIGURE ..... **IV**

CHAPTER1- INTRODUCTION **1**

1.1 INTRODUCTION: 1

1.2 OVERVIEW: 1

1.3 PROBLEM STATEMENTS: 2

1.4 OBJECTIVE 2

1.5 SCOPE OF PROJECT: 3

1.5.1 Project Scope: *3*

1.5.2 User Scope: *3*

1.6 PROJECT JUSTIFICATION: 3

1.7 TIMING – GANTT CHART 4

..... **4**

1.7.1 Task1- Find supervisor *4*

1.7.2 Task2- Project proposal *4*

1.7.3 Task3- Planning & Research *4*

1.7.4 Task4- Chapter1-Introduction *4*

1.7.5 Task5- Chapter2-literature view *5*

1.7.6 Task6- Chapter3-Methodology *5*

1.7.7 Task7- Chapter4-- requirements analysis *5*

1.7.8 Task8- Chapter5-- Design *5*

UNIFIED MODEL LANGUAGE (UML) TO DESIGN FLOW CHART, CASE DIAGRAM, CONTEXT DIAGRAM .. 5

1.7.9 Task9- Poster presentation *5*

1.7.10 Task10- implementation *5*

1.7.11 Task11- Testing & evaluation *5*

1.7.12 Task12- Video running application *6*

1.7.13 Task13- Final report *6*

1.7.14 Task14- Oral presentation *6*

2 CHAPTER2: LITERATURE REVIEW **7**

2.1 INTRODUCTION 7

2.2 BACKGROUND OF VISA DEPARTMENT 7

2.3 QUEUING 8

2.4 ANDROID STUDIO PLATFORMS 9

2.5 FIREBASE REALTIME DATABASE 9

2.6 RELATED APPLICATION 10

3 CHAPTER3- METHODOLOGY **11**

3.1 INTRODUCTION 11

3.2 RESEARCH METHODOLOGY 11

3.3 AGILE 11

3.4 SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC): 12

3.4.1 Phase1- Planning *14*

3.4.2 Phase2 – Analysis *15*

3.4.3 Phase3 – Design *15*

3.4.4 Phase4 – Implementation *15*

3.4.5 Phase5 – Testing *16*

3.5 HARDWARE AND SOFTWARE REQUIREMENTS 16

3.5.1 Introduction *16*

3.5.2 Hardware requirements *16*

3.5.3 Software requirements *17*

4 CHAPTER 4: DOCUMENTATION REQUIREMENTS **18**

4.1 REQUIREMENT GATHERING TECHNIQUES 18

4.2 SURVEY METHOD 19

4.2.1 Objectives of survey *19*

4.3 ANALYSIS REQUIREMENT 19

4.4 FUNCTIONAL REQUIREMENTS 26

4.4.1 Input requirements *26*

4.4.2 Operational requirements *26*

4.4.3 Output operational requirements *26*

4.4.4 Performance *26*

4.4.5 Maintenance *26*

4.5 NON-FUNCTIONAL REQUIREMENTS 27

4.5.1 Usability *27*

4.5.2 Feasibility *27*

4.5.3 Economic feasibility *27*

5 CHAPTER 5-DESIGN 28

5.1 USE CASE DIAGRAM: 28

5.2 CONTEXT DIAGRAM: 29

5.3 STORYBOARD: 30

5.4 FLOWCHART: 31

5.5 ENTITY RELATIONSHIP DIAGRAM(ERD): 32

5.6 DATAFLOW DIAGRAM: 33

6 CHAPTER 6: TESTING 34

6.1 INTRODUCTION 34

7 CHAPTER 7-IMPLEMENTATION 40

7.1 INTRODUCTION: 40

7.1.1 Language used: *40*

7.1.2 Extensible Markup Language (XML) *44*

7.1.3 Firebase Database *45*

7.2 SOFTWARE USED: 46

7.2.1 Android Studio: *46*

7.3 SCREEN SHOTS OF THE APPLICATION: 46

7.3.1 LogIn Functionality *46*

7.3.2 SignUp Functionality *47*

7.3.3 Main page Functionality: *48*

7.3.4 Get Number Functionality: *49*

7.3.5 Collect/submit Functionality: *50*

7.3.6 View info Functionality: *51*

7.4 DATABASE STRUCTURE 52

7.4.1 Introduction: *52*

7.4.2 Entity Relationship Diagram (ERD): *52*

7.4.3 Data Dictionary: *53*

8 CHAPTER 8-CONCLUSION 55

9 REFERENCE 56

10 APPENDIX 57

CHAPTER1: INTRODUCTION

1.1 Introduction:

Queues (or waiting lines) are seen in everyday life like in the waiting lists in the hospitals, gas stations. Queues are formed when Students demanding service are more or r the facility does not work efficiently or takes more than the time prescribed to service a Student. Current system doesn't cover student requirements, this system include physical queues, Students gather in a small lobby which uncomfortable for students, this could also worsen, the atmosphere for students who are currently receiving service.

1.2 Overview:

I developed an android application that will allow Students to join a virtual queue via mobile phone, and allowing them to check there turn without the need to be present at the visa hall also, providing an estimated time that gives the Students proper idea when their turn will start, this application will used in visa department in Limkokwing university.

1.3 Problem statements:

Wasting time while waiting for long time on the queue for reserving number and waiting to get served are inefficient way to operate day to day operation with the students, also students uncomfortable because they are waiting in small place and to crowded with a lot of student which make students dissatisfied with. Lack of understanding in calling number which make manually reserving numbers for specific issue hard to determent by the individual user. Increasing registered students rate last two years which make visa department unable to handle the high demand regarding on managing and serving student needs, this problem makes some students leave without being served and make them angry because they wasting their time.

1.4 Objective

- to research the current system to managing waiting students.
- to provide a better overall system to the student.
- to reduce wasting time, by show estimate waiting time and sent notification when their number is nearing.
- to provide an understanding in calling numbers, and improve the handled of high demand regarding on managing and serving student needs.

1.5 Scope of project:

1.5.1 Project Scope:

The Queue Numbering System is designed to manage number of students with single department and multiple counters, there are two operation submit, collect and four counters in this system.

1.5.2 User Scope:

The Queue Numbering System is designed to meet the needs of students and provide,

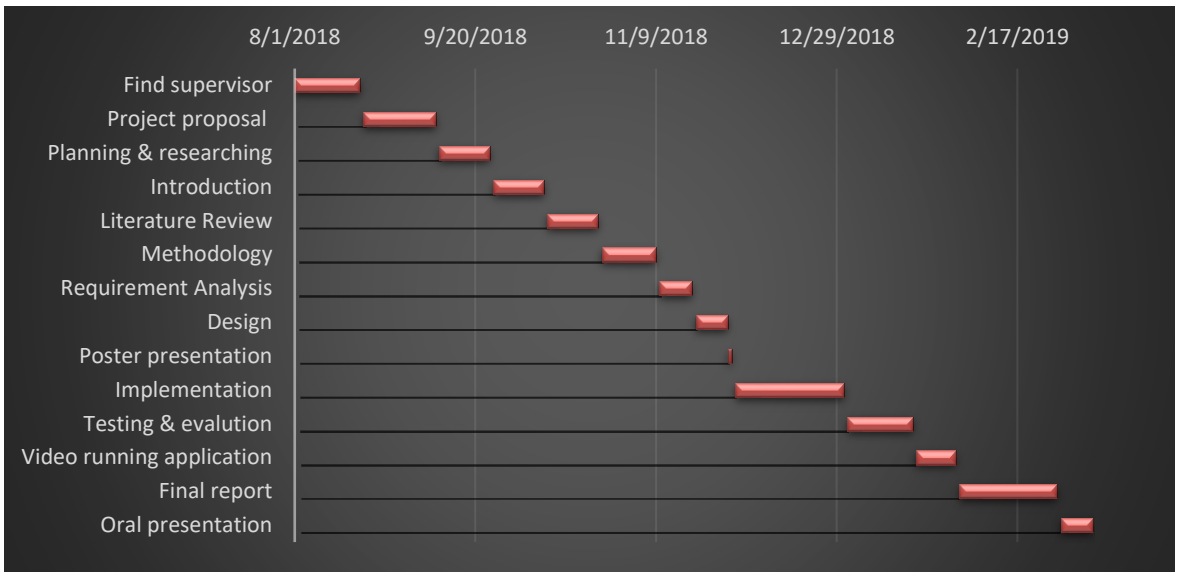
Useful environment that makes students able to do any of the operations too easily, and

Makes the student comfortable and satisfying.

1.6 Project Justification:

This system created to help student to get their queue number by using queueing numbering application, this free application is able to download from android apps store, students can provide their feedback, and those feedbacks can make as improvements in future. Students can go anywhere that mobile internet service is available while their waiting, when student's queue number is nearing, student will receive a notification as remind. Student can direct go counter to show their queue number and get services. Based on their queue number, counter staff can get Student's details and needs and, provide services in shortest time.

1.7 Timing – Gantt chart



1.7.1 Task1- Find supervisor

In first of august selecting the best supervisor. Finding process based on the most qualified lectures at faculty of information communication technology. Ms. Suhaila was the selected to propose my project idea.

1.7.2 Task2- Project proposal

In this task I submit project proposal to Ms. Suhaila, discussion all the major point, problem statement and objective, Ms. Suhaila accept the project idea then the supervisor forms have been submitted

1.7.3 Task3- Planning & Research

Intensive proper research to develop complete based on the problem identified in the first chapter, and to meet the objective of this project.

1.7.4 Task4- Chapter1-Introduction

At the beginning of September, introduction of the project has finished. Which make the concept of project clearer to start searching and looking for related system to find out drawbacks of each project and how this project will be complete and fix all the error which has found in previous system

1.7.5 Task5- Chapter2-literature view

at the end of September, literature view has completed, organization background managed to be implemented in Limkokwing university campus.

1.7.6 Task6- Chapter3-Methodology

Identify method tools, agile methodology has been adopted, five phases have been selected to go through the development process, case diagram and flow clearly explained in detail.

1.7.7 Task7- Chapter4-- requirements analysis

Adopt Survey method with Student in Limkokwing university to get some extra information on the upcoming application

1.7.8 Task8- Chapter5-- Design

Unified model language (UML) to design flow chart, case diagram, context diagram.

1.7.9 Task9- Poster presentation

Design a poster to present my idea in front of my respectable faculty lectures, and get a feedback that may help in development process.

1.7.10 Task10- implementation

In this task coding will start and building structure of project, implementing all the functions in application.

1.7.11 Task11- Testing & evaluation

Testing process will consist of multiple stage, unit testing, and user testing will be consider to test the effective of function of the application.

1.7.12 Task12- Video running application

Capture video to show the audience, lectures and supervisor how the function of application works.

1.7.13 Task13- Final report

Submitting final report to the supervisor Ms. Suhaila.

1.7.14 Task14- Oral presentation

Present the final presentation of the project in front respectable lectures.

2 CHAPTER2: LITERATURE REVIEW

2.1 Introduction

This application will develop using android studio, object-oriented programming, and fire database. I will talk about android related works, also I will discuss about using of object-oriented Programming, and I will give over view about database I use it in this application.

2.2 Background of visa department

Queuing in visa department is an approach that lining up of students in visa hall in order to be served by Visa department employees at each counter. Students have to stay inside the visa department hall until there turn come without knowing time needed to be spent, some students even leave visa hall before being served, This and other obstruction result too much delay in students waiting time.

2.3 Queuing

A queue is a linear list in which data is inserted at one end, Queues are used to help divide subtasks. The queue system is mainly used for organizing applications submissions so when a student approaches the visa department, they will present their purpose and be served with a queuing number accordingly after that they will wait for the number to show up on a screen and then they will be served based on the counter representing that number. The number will begiven based on availability and that number is linked to a screen that shows the circulation of the numbers in the queuing system every number taken will show on the screen by detecting the missing number from the database of the system and presenting it accordingly.

.

2.4 Android studio platforms

Android studio is an integrated development environment (IDE) based on IntelliJ IDEA, it's powerful code editor and development tools, android studio offer a lot of feature to enhance your coding technique when make application, many features available such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- Instant Run to push changes to your running app without building a new APK

In contrast with its major competitor's application market, Apple's App Store which rejects application submissions that are deemed too simple or are similar to existing apps. Android Marketplace doesn't censor its apps. In other words, apps developed by any developers will have equal chances to enter and to be published in the market

2.5 Firebase Realtime Database

Store and sync data with our NoSQL cloud database. Data is synced across all clients in Realtime, and remains available when your app goes offline. The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and in Realtime to every connected client. When you build cross-platform apps with, IOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

2.6 Related application

This application **Q-smart Queue Management System Terminal** is same what I am going to develop and I will make some changes, to make sure that the application will be useful for students and friendly user interface. I will add some information related to visa department can help the students such as Renew visa, reapply visa, transfer visa, and ask visa department for special issues.

Figure 1sample on related application

3 CHAPTER3- METHODOLOGY

3.1 *Introduction*

Visa department always suffers from waiting line problem due to the presents of student, in visa department hall until they get served. The new system has been designed in such a way that it limits number of students inside the visa department hall and giving them the opportunity, to benefit their time without worries missing their turn.

3.2 *Research Methodology*

Agile will be the main method used in this research paper, the reason of using this method due to agile method enables to enhance collaboration between system developer and application users. The benefit of using agile method it allows the developer to retrieve feedback from application user and notify the application user the changing requirements has been changed.

3.3 *Agile*

Agile is a hype word that the IT industry use to describe an alternative method of project Management. An agile is a very empowering process that helps companies design and build, the right product. he management process is very beneficial for software companies because it helps them analyze and improve their product throughout its development. This enables companies to produce a highly valuable product.

3.4 *Software Development Life Cycle (SDLC):*

is a process used by the software industry to design, develop and test high quality softwares. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process. Software development life cycle is a series of three main phases planning, design, and implementation.

.

Planning	**1.** Identify the project idea. **2.** Identify the problem statement. **3.** Select a proper solution. **4.** Identify objectives.
Analysis	**1.** Searching previous research. **2.** Gathering information analyze- questionnaire.
Design	**1.** Use Case diagram. **2.** Flowchart **3.** Entity relationship diagram (ERD). **4.** Data Flow Diagram (DFD). **5.** Context diagram. **6.** Story board.
Implementation	**1.** Java programming language. **2.** Android platform.
Testing	**1.** GUI testing. **2.** Integration testing. **3.** Unit testing **4.** User testing.

Table 1SDLC Phases summary

3.4.1 Phase1- Planning

In the first phase students and visitors faced daily problem in visa department affecting the daily process. waiting and wasted time has been clearly determine, find out the proper solution as defined in the first chapter. Objective and problems where evaluated to produce a propriate solution supposed to solve the problem found in visa department. This stage is very important because it describe the project building process and how the project going to implement. Queuing Numbering System project must identify the benefit for universities and collage. This project will use same technique exist in previous research by adding new idea which focusing in solving the problem faced the students and visitors. Eventually sequence in developing this project (time frame) specifying which task must be done within the project life cycle.

3.4.2 Phase2 – Analysis

Start with this phase, previous research and related system were selected to use it as approach during this project (refer to chapter2). Alongside with questionnaire include 3 sections ,13 questions and 20 respondents. result which has been collected from students, gathering and analysis is very affective technique to get wider and better understanding on how the system serve particular problem.

3.4.3 Phase3 – Design

All the requirements define in analysis and planning phase will translated into structure and specification, all the diagrams in unified modeling language (UML) such as use case diagram and entity relationship diagrams will draw in this phase considering the time frame and requirements in analysis phase.

3.4.4 Phase4 – Implementation

Considering as most important phase coding and development will start, the requirements in design phase identified within UML diagrams are going to be coded using java programming language as code language for developing android application. To deliver this project android studio IDE play the main role while development process. as known as implementation phase is very risky because error and bugs may face while developing this project as long as using agile methodology it’s provide the ability to modified some concept in design phase. Agile support the project to avoid errors and deliver complete application that follow the requirements

3.4.5 Phase5 – Testing

First beta version will publish for the users to make testing and evaluation, the purpose of testing is to know if the system functionality working as needed and feasibility of project is fit for students and users, the main testing approach will used for testing purpose are unit testing, GUI testing, integration testing and user testing. Group of students from different faculties will be selected to test and evaluate this project.

3.5 Hardware and software requirements

3.5.1 Introduction

Any project needs hardware and software requirements to develop or create, these requirements are considered the main part of projects structure. To create queuing numbering system hardware and software requirements must be identified clearly as following:

3.5.2 Hardware requirements

In the development of this project, certain hardware requirements and specifications were needed in order for application to be functional. This application is developed for android OS platform-based phones and tablets. The following are the hardware specifications:

- The device should have at least 100MB of free space.
- The device must have 1GB RAM.
- The device must have at least android version 4.0.
- The device must have Wi-Fi or 3G data.
- The device should have multi touch screen.

3.5.3 Software requirements

The Queuing Numbering System project needs will determine specifications that must software environment needs to complete the project. The following are the main software requirements:

- **Java programing language.**

  Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, I will use this language to write the code of my project.

- **Android studio SDK version 3.2.1.**

  This was needed to build and compile the code that I made and make it APK packages for android phone installation.

- **Firebase real time database.**

  We use this to store the data and because its more efficient and data stored synchronized in Realtime to every connected client.

4 CHAPTER 4: DOCUMENTATION REQUIREMENTS

4.1 Requirement Gathering Techniques

For every project especially, those which involves development of applications and systems, gathering the right requirements during the analysis phase of the SDLC is highly required. In fact, every developer especially senior developers and team leaders, should at least possess the basic skills of gathering good requirements for their project. If requirements are unclear, or undefined at the earlier stage, risks of redoing the whole project will be high which eventually cause wastage of time, financial resources and energy. Plus, quality of the final deliverable may be badly affected too. Every developer should bear in mind that the final solution of the project should meet the targeted user needs. A project will still be deemed as bad or of low quality no matter how attractive or high-tech the system is if requirements of users are not satisfied There are many approaches which are commonly used to gather user requirements. Survey method involves use of electronic or paper-based medium to garner needs and requirements from clients or stakeholders. This method is best used when there is a need to gather information from up to hundreds to thousands of stakeholders and time is a limiting factor. Plus, with the use of survey, developers or team leaders may retrieve focused data efficiently and validate assumptions effectively. In the process of conducting a survey, little scheduling effort and stakeholder time and commitment are required, thus making the process an easy-going process. For data which are required to be tabulated to gain statistical preferences information, survey is highly suitable to be used. However, survey method does have its cons like any other techniques. If the survey is made to be very large, responses from participants may most likely to be low. Questions being asked in the survey may also reflect developer's preconceived ideas or assumptions, thus reducing the chance of getting information on what user really need.

4.2 Survey method

Using of Queuing Numbering System by students and public, a lot of students are available to be participant in this survey. This technique considers as best way to gather the information required. I can gather information about student's expectation on the project I am currently developing. The participant 25.

4.2.1 Objectives of survey

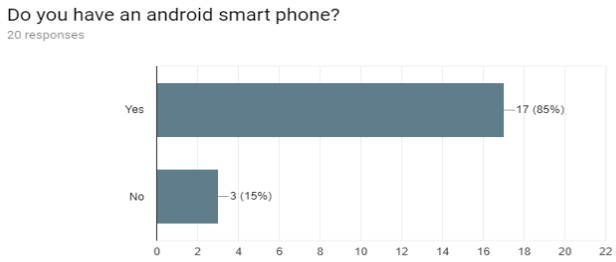
- To know the percentage of students struggling while waiting at visa department.
- To determent how frequently the student visit visa department.
- To count number of students using android operation system.
- To collect student opinions and their suggestion for the project.

4.3 Analysis requirement

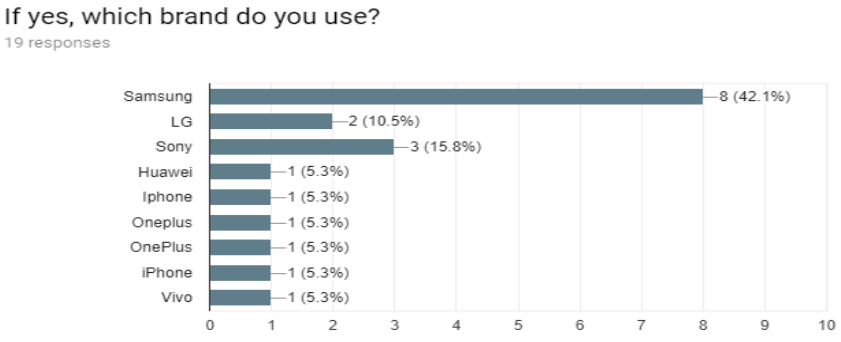
As one of techniques to have a better understanding on the requirements need for project to study student opinion and gathering information from them. This survey has been procced. There were up to 20 respondents done this survey.

Result of section 1 – Android smart phone users among Limkokwing university.

In section 1 of the survey, questions which define the popularity of android phone users among Limkokwing students were asked. The following chart shows the number of students who use Android smart phone and those who don't.

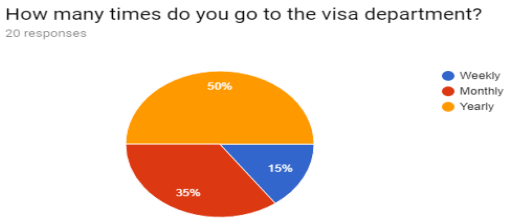


- From the chart above, we can understand that 85% of the 20 respondents they have at least A device running on android operating system whether phone or tablet. The others 15% They did not use android devices.
- From the result shown. It can be efficient to implement the project to run on android Operating system, with the fact of popular manufacture companies like Samsung, LG, Sony and many other devices we found as shown below:



Result of section 2 – study on existing service

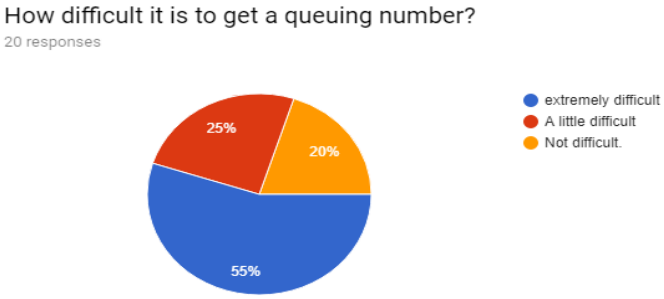
In this section of the survey, questions were asked to gain information from students frequency visiting visa department in Limkokwing university, which service they had in visa department this information's were collected in section two. The following chart present result on how many times usually students went to visa department.



From above chart we can see the percentage of student the went to visa department Weekly, monthly, and yearly.

- From chart above we see there were 50% (10 students) usually visiting visa department Yearly.
- 35% (7 students) they usually visiting visa department monthly.
- 15% (3 students) the usually visiting visa department weekly.

Most of respondents in the following chart they were faced difficulty in getting a queuing number in visa department.



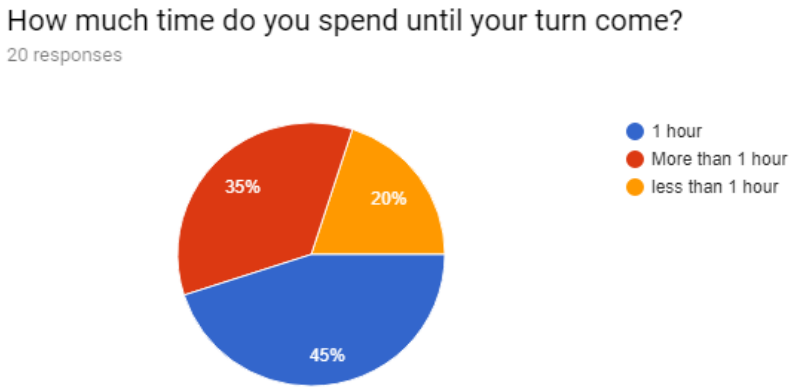
Justification:

from the result that shown in the chart above, we can say that biggest percentage of students struggled to getting a queuing number which mean with implementing this project will make

getting a queuing number easier than before.

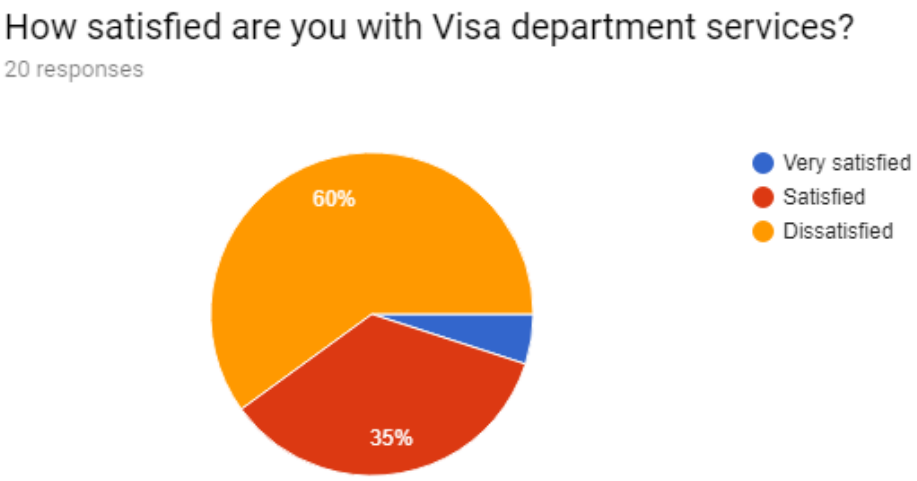
- 11 students (55% responses) they found that getting a queuing number in visa department Is extremely difficult.
- 5 students (25% of responses) they said that getting a queuing number in visa department is a Little difficult.
- 4 students (20% of responses) they didn't face any difficulty to getting a queuing number in visa department.

Students were asked how long they waited until there turn came, the following chart shows the result:



- 9 students (45% of responses) said that they had to wait an hour in visa department until there turn came and getting served.
- 7 students (35% of responses) said that they had to wait more than one hour in visa department until there turn came and getting served.
- 4 students (20% of responses) said that they had to wait less than one hour in visa department until there turn came and getting served.

In second section also, students were asked about if they were satisfied with visa department services and the most said they were dissatisfied as the result shown in the following chart.



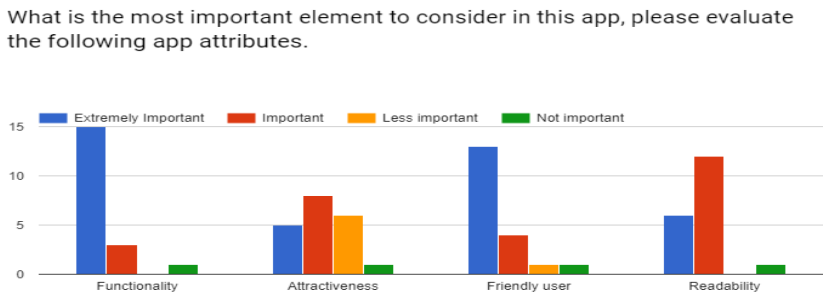
- From this chart above we can see that the biggest percentage of dissatisfied students,

12 student (60% of responses) they were dissatisfied about visa department services and they waiting too long time until they get served.

- 7 students (35% of responses) were satisfied about visa department services.
- 1 student (5% of responses) was very satisfied with visa department services.

Result of section 3 - Study of expectations

The following chart explain respondent's towards application based on friendly user, Attractiveness, Functionality, and readability.



- As showing in above chart, conclusion out of 20 respondents can be made.
- Functionality, Friendly user were the most important element for respondents to be exist in the upcoming application.
- Readability, and attractiveness were not the highest priority for respondent in upcoming application.
- 15 students said functionality is the most important element should consider while developing this application.
- 13 students said friendly user is important element in this application.
- As there are different ratings from the respondents, all the elements will be considered the highest evaluated feature, and will be more focus while developing.

Finally, here is the selected suggestion which are received when the survey is procced.

- Easy to use by students.
- Easy to manage.
- Avoid system crashes scenarios.

4.4 *Functional requirements*

In this section I will describe the functional requirements.

4.4.1 Input requirements

This application is easy to access by users or admin ID and password and Login criteria only for admin. User can contact this visa department administration by fill up form designed in application interface.

4.4.2 Operational requirements

in this phase ensure the functions of the applications and connectivity. Check database connection, check network connection.

4.4.3 Output operational requirements

In this section analysis the data returned form database and ensure the right logic and result in a special format representation.

4.4.4 Performance

Ensure the performance is smooth and fast enough.

4.4.5 Maintenance

Structure of system enable developer to edit in maintain fix the errors in the application.

4.5 *Non-functional requirements*

4.5.1 Usability

Ensure the system is easy to use by students as well as the admin.

4.5.2 Feasibility

Ensure quality of the attractiveness provided to user and admin.

4.5.3 Economic feasibility

The total cost of project will meet the expected cost in planning and analysis phase.

5 CHAPTER 5-DESIGN

5.1 *Use case diagram:*

This diagram used to describe a set of actions in the system as show below in the diagram, there are two actors the admin and the user both of them have certain actions. The admin can log in to the application after log in he can call the number which means the queue number, also can collect forms from student and answer inquries. User can get number whether for collect passport or submit passport,can view information about the doucuments have to submit, and can ask about inquires whether were Frequent Question(F&Q) or ask for special issue.

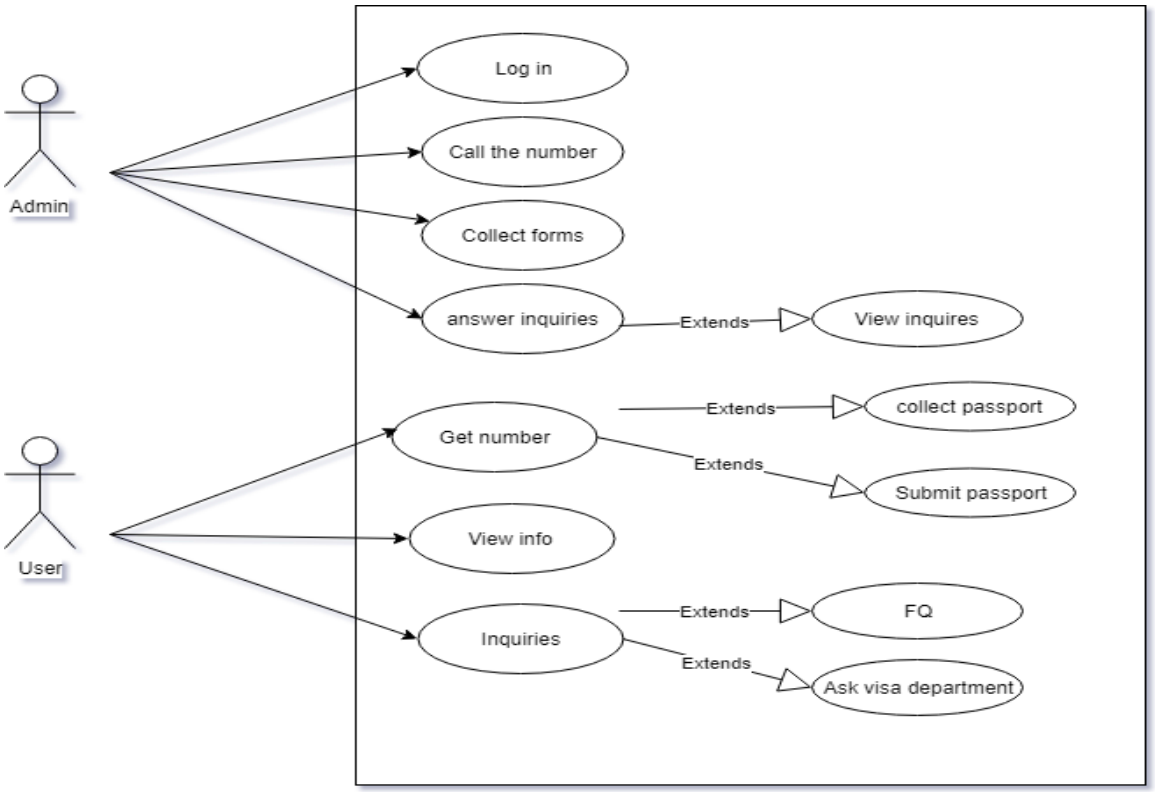


Figure 2Use case diagram

5.2 *Context Diagram:*

This diagram show the first level of data flow and the relationship with extenel entities. There are User, Operating System(OS), and Admin. We see here how the OS reactes with admin and user, user ask For queuing number and system return for him the queuing number, user submit query form to the system and system send it to the list of query forms for the admin and as soon as the admin answer the system will update query form for user. Admin logIn to the system system check information if correct returen to admin LogIn successfully admin update queuing number and send it to the system then user know how many students still need to be served untill his turn come, and admin see the list of query forms and confirm query forms and sendt to the system then system make update about query form for the user.

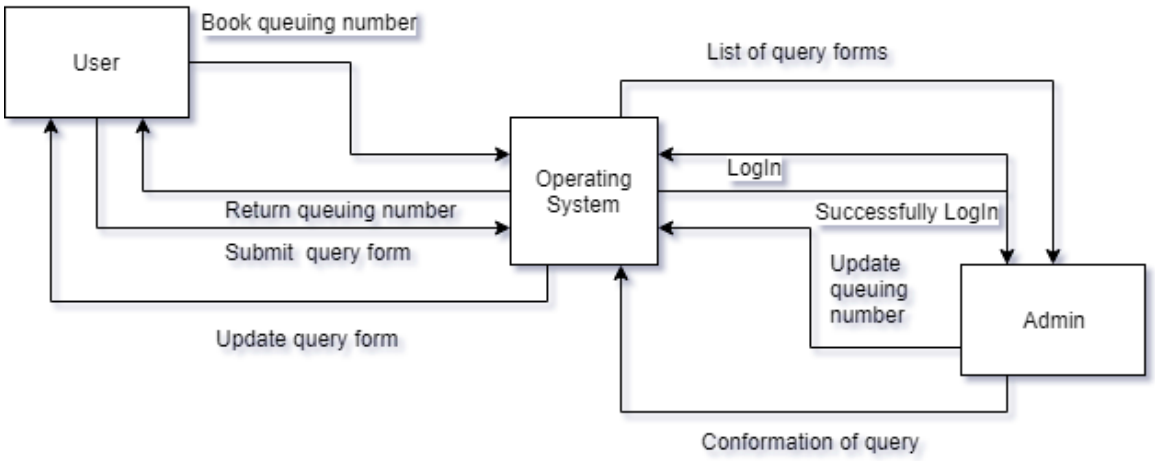


Figure 3Context diagram

5.3 *Storyboard:*

the following storyboard represent the main page in the application with other functionality, First we have main page which shows when we start the application in this page we to choices admin or user.When we click on admin we will go to admin page which has Employee_ID,Employee_password after filling the ID and Password we have login after we prees log in it takes us to another page which has call number, next number, and answer inquires. For the user if we press user it takes us to user page which has get number, view info, and inquires. If we press get number will take us to page have two options either for collect passport or submit passport, for view info will show a display has information about the documents you have to submit first, and for inquieres will take you to page has two options one for F&Q and the other for asking the visa department.

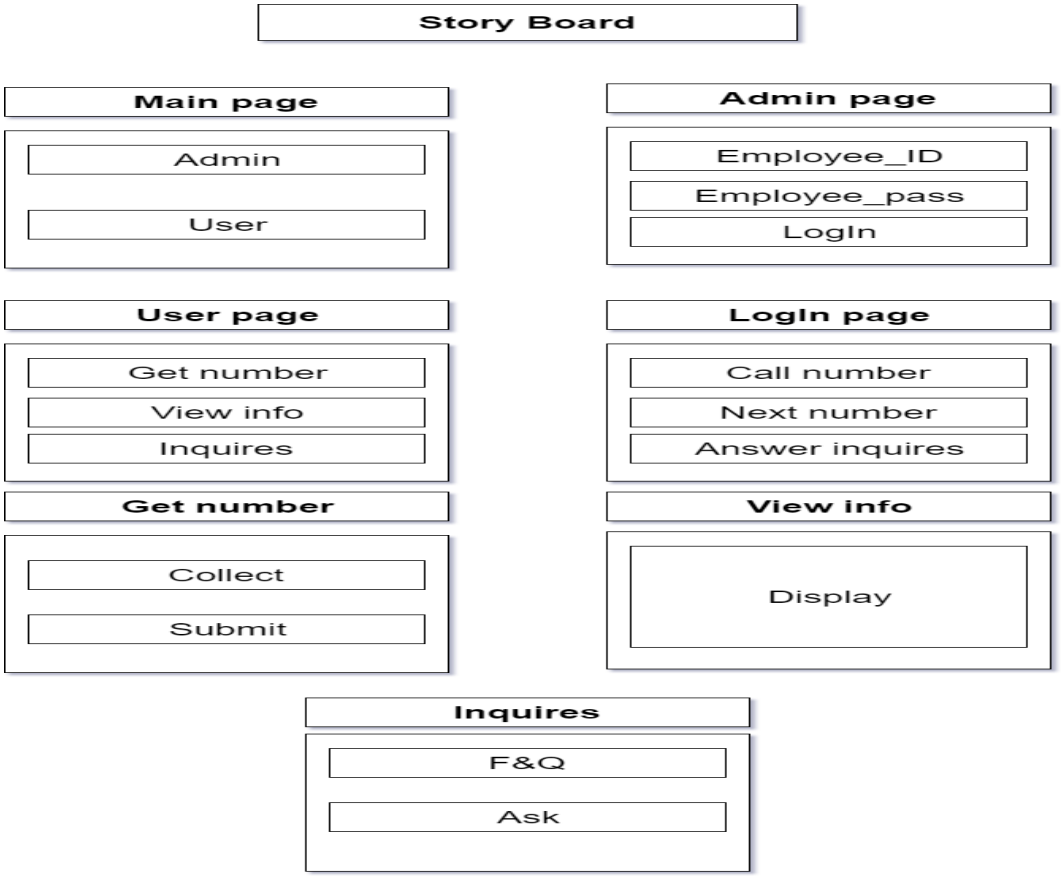


Figure 4Story Board

5.4 *Flowchart:*

The diagram below represent work flow or process of the application this process for the user start application then choose place number or inquires for place number have two options one for collect passport and the other one for submit passport after choosing one of the options then the application Will show the number and go to the end. For inquires after inquires you ask visa department then go the end of the application.

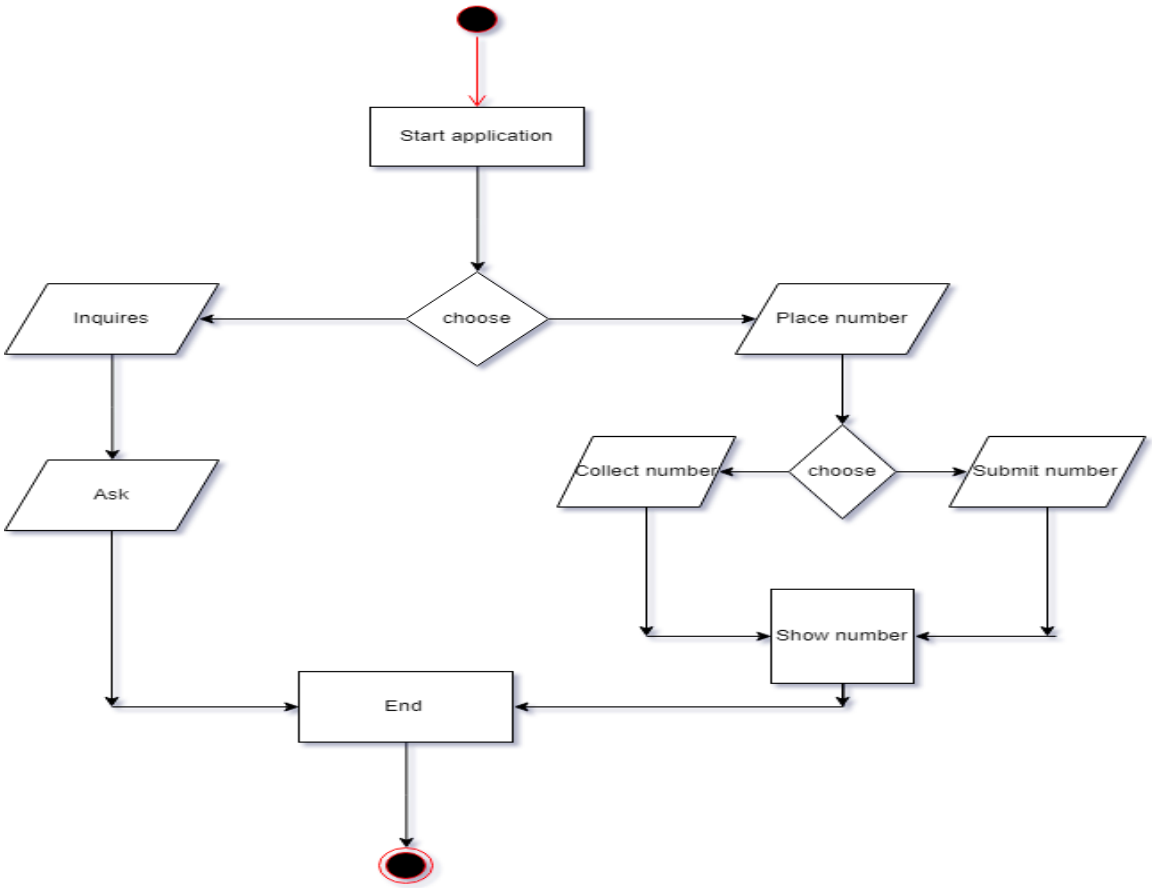


Figure 5Flowchart diagram

5.5 *Entity Relationship Diagram(ERD):*

This diagram below show the relationship between sets of entities stored in database, we have three Tables Ticket table, User table and inquire. The relationship between ticket table and user is one to one,And betwwen user tabel and inquire the relationship is many to many.

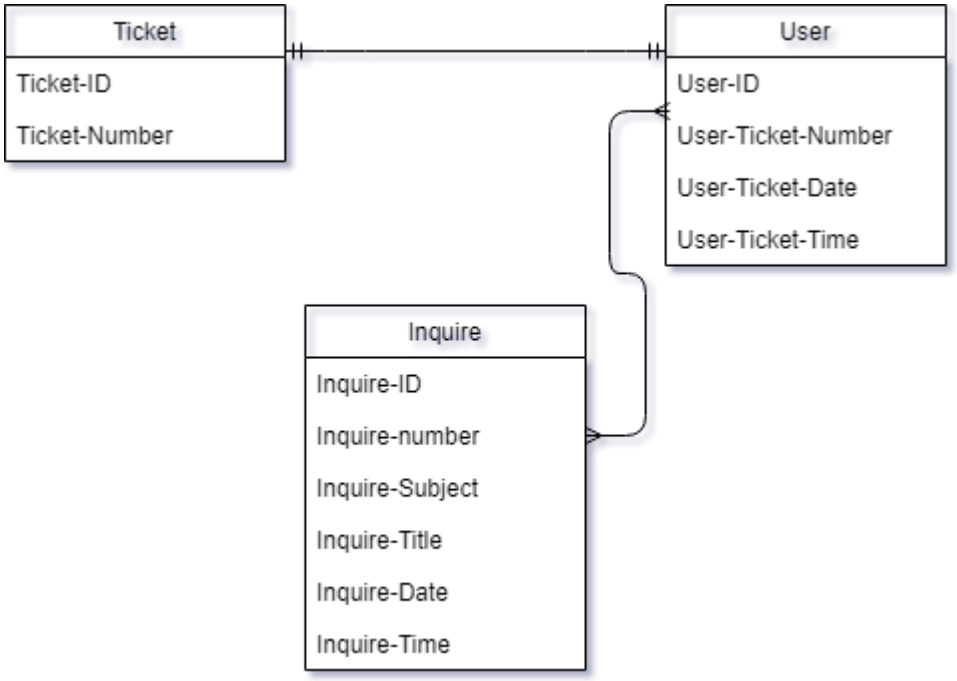


Figure 6Entity Relationship Diagram

5.6 *Dataflow Diagram:*

This diagram more detailed than DFD level 0 in this diagram I explain what data flow for each process for Login the data flow is ID & password Details after Login there is call number and the data flow is the Ticket ID, collect forms the data flow is Ticket ID & student ID, and the data flow from collect forms to Answer inquires is quire Details this for the admin, for the user to get number the data flow is student Details, and view rules/info the data flow is process details.

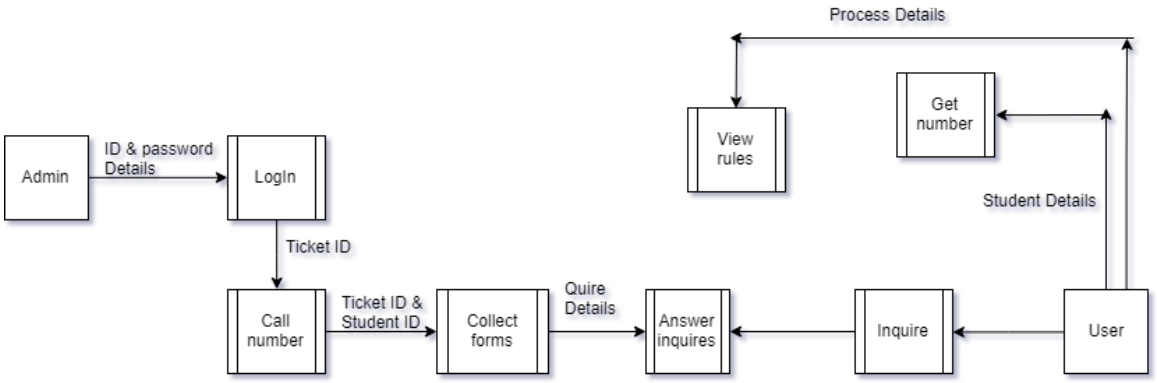


Figure 7Dataflow Diagram

6 CHAPTER 6: TESTING

6.1 introduction

In the testing phase, a few evaluators are chosen to test the Queuing Number for errors being deployed. The following table shows the test plan which includes the expected results and actual results.

Test case id	Test case name	Test case desc	Test steps		
			Step	Expected	Test status (P/F)
Login 01	Validate Login	To verify login name and password are correct	Enter login name and password and click login button	An error message “Login name or password incorrect	
SignUp 01	Validate SignUp	To verify SignUp information are fill name, email address, and password	Fill the content name, email address, and password and click SignUp button	An error message “there is and empty filed you have to fill it.	
Collect number 01	Validate collect number generator	To verify generating collect number is working	Press the collect number button	Show collected number.	
Submit number 01	Validate submit number generator	To verify generating submit number is working	Press the submit number button	Show submit number	
View info 01	Validate View information records	To verify view info is showing the students records	Press the View info button	Show the student records name, and ticket number	

6.2 Unit testing:

Pre-conditions: Developer performs unit testing.

- Developer should know the source code of each unit.
- Mobile phone should be in working condition.

Dependencies:

- · Source code of each unit
- · Mobile phone manual

- unit testing, developer should have to test different units of mobile phone. Mobile phone consists of following
- units-
- 1) Keypad 2) Messages 3) Contacts 4) Gallery
- 5) Play Music 6) Alarm 7) Settings
- Test cases required for testing each unit are as follows:

Step no	Test cases	Test data	Expected result	Actual result	Status (p/f)
1	**Unit-Keypad** Check that whether All keys (alphabets A to Z ,a to z, 0 to 9, special symbols) Functional keys like space, enter(next line) key, delete key, calling button(Green), call rejection button(red) Working correctly or not.	Mobile Phone	All keys (alphabets A to Z ,a to z, 0 to 9, special symbols) Functional keys like space, enter(next line) key, delete key, calling button(Green), call rejection button(red) should Work correctly.	All keys functioning properly.	P
2.1	**Unit-Messaging-** Check that user can type a message.	Mobile phone	User should be able to press all alphanumeric keys. All alpha-numeric keys, enter and delete key should work properly.	All keys functioning properly. User can type message.	P

2.2	Check that user can send a message	Mobile phone, available network signal, balance amount	After typing a message user should be able to send it by pressing send button.	Message sent successfully.	P
2.3	Check that user can receive a message.	Mobile phone, available network signal	User should be able to receive a message and saved it to inbox.	Message received successfully.	P
3.1	**Unit- Contacts And Calling** Check that user can **establish (dial) a call.**	Mobile phone Contact no., Network signal, balance amount	User should be able to dial contact no. by dialing different nos.(0 to 9) and make a call by pressing caller button(usually green coloured)	Contact no. dialed and call established.	P
3.2	Check that user can **accept call .**	Mobile phone, available network signal	Mobile phone should be able to play a ringtone and vibrate. User should be able to accept call by pressing call acceptance (green button)	Mobile phone ringed and vibrated. User accepted call successfully.	p
3.3	Check that user can **reject the call**.	Mobile Phone, available network signal	Mobile phone should be able to play a ringtone and vibrate. User should be able to reject call by pressing call rejection (red button).	Mobile phone ringed and vibrated. User rejected call Successfully	P
3.4	Check that user can **Save a contact no**.	Mobile phone Contact no.,name Memory.	User can enter contact no. by using keypad and should get saved into memory.	Contact saved successfully.	P
3.5	Check that user can **modify a contact no.**	Mobile phone Contact no.,name Memory.	User can enter new contact no. by using keypad and should get saved into memory.	Contact saved Successfully	P

3.6	Check that user can **delete a contact no.**	Mobile phone Contact no.,name Memory.	User can enter contact name. by using keypad and should get deleted from memory.	Contact deleted successfully.	P
3.7	Check that user should not create **duplicate contact.**	Mobile phone Contact no.,name Memory.	User should enter contact name by using keypad and if he enters duplicate name mobile should notify user and save that no. with alternate name.	Duplicate contact should not be made. Alternate contact name should be created.	p

6.3 Integration testing

Pre-conditions:

· All units must be tested before integration testing.
· Developer should integrate all the units into a single system.
· Mobile phone should be in working condition.

Dependencies:

· Integrated system
· Mobile phone manual

Step no	Test cases	Test data	Expected result	Actual result	Status (p/f)
1	Check that keypad works while typing a message	Mobile Phone	User should be able to type a message by using different keys on keypad.	Message typed successfully.	P
2	Check that user can type a message while listening to song	Mobile Phone	User should be able to type a message by using different keys on keypad while listening to songs	Message typed Successfully	P
3	Check that user use a keypad while he is talking on call.	Mobile Phone	User should be able to type while he is on call.	Call made successfully.	P
4	Check whether speakers and	Mobile Phone	Speakers should function well so that	During calling speakers and	P

	microphone working or not while calling.		user can listen to receivers voice. Microphone should be able to capture audio signals(dialers voice) for proper communication.	microphone working properly	
5	Check that while charging a phone user should be able to perform all phone functions(contacting, Messaging,playing songs,etc)	Mobile Phone,charger, Power supply	User should be able to access different functions of phone while charging it.	User can access phone while charging it.	P
6	Check that user can send images, audio,videos as MMS(Multi Media Message Service).	Mobile Phone	User should be able to send songs and images	User can send Images and songs successfully	P
7	Check that alarm and phone timing working in synchronisation.	Mobile Phone	Phone and alarm timing shoud work synchronise way.	Phone and alarm timing synchronised.	p

6.4 System testing

Pre-conditions:

· Integration testing.
· All units should be integrated as **one entire system.**
· Mobile phone should be in working condition.

Dependencies:

· Entire system
· SRS document
· Mobile phone manual

Test no	Test cases	Test data	Expected result	Actual result	Status (p/f)
1	Check the mobile phone as entire system	Mobile phone,network signal	Diffrerent functions of mobile phone tested.	Phone functions successfully.	P
2	Check that user can dial and receive a call.	Mobile phone, network signal	user should be able to dial and receive a call	Call made successfully.	P
3	Check that user can type ,send and receive a message.	Mobile phone, network signal	user should be able to type ,send and receive a message.	Message sent successfully.	P
4	Check that user can access images, play songs.	Mobile phone	user should be able to access images, play songs.	user can access images, play songs successfully	P
5	Check that user can set and cancel alarm on required time or not.	Mobile phone	user should be able to set and cancel alarm on required time or not.	Alarm set or cancelled successfully	p

7 CHAPTER 7-IMPLEMENTATION

7.1 Introduction*:*

This Chapter include description on what software and language are used for the development of Queuing Number application. The project implements java programming language, XML and Firebase Database. The project will be capable of running on any android device.

7.1.1 Language used:

In the development of Queuing number application, the following programing languages are used:

7.1.1.1 Java Programing Language

Java language is the main programming language supported by Android. Java concepts such as such as inheritance, usage of super class and subclass and abstract and java syntaxes are applied throughout the development of the Queuing Number application (Using android studio software).

7.1.1.2 Application code using java

Login class:

This java code will retrieve data stored in Firebase database, data about user so user can login successfully also there is validation if user write something incorrect will never login.

```
public class LoginActivity extends AppCompatActivity {
    TextView mCreateAccount;
    EditText mEmail, mPass;
    Button mLogin;
    FirebaseAuth mAuth;
    FirebaseUser mUser;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        mCreateAccount = findViewById(R.id.create);
        mEmail = findViewById(R.id.email);
        mPass = findViewById(R.id.password);
        mLogin = findViewById(R.id.login);
        mAuth  = FirebaseAuth.getInstance();
        mCreateAccount.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent( packageContext: LoginActivity.this , SignupActivity.class)); }});
        mLogin.setOnClickListener((v) -> {
                if (mEmail.getText().toString().isEmpty() || mPass.getText().toString().isEmpty()){
                    Toast.makeText( context: LoginActivity.this,  text: "Please enter email and password", Toast.LENGTH_SHORT).show();
                }else {
                    Toast.makeText( context: LoginActivity.this,  text: "Logging in", Toast.LENGTH_SHORT).show();
                    mAuth.signInWithEmailAndPassword(mEmail.getText().toString() , mPass.getText().toString()).addOnCompleteListener((task) ->
                            if (task.isSuccessful()){
                                userLoggedIn();
                            }else {
                                Toast.makeText( context: LoginActivity.this, task.getException().toString(), Toast.LENGTH_SHORT).show();
                            }
                    });
```

Figure 8 LogIn Activity

Signup class:

The purpose of this code is to save data about the user in firebase database, so when user login the log in class retrieve the data stored in firebase database.

```
public class SignupActivity extends AppCompatActivity {

    EditText mName , mEmail , mPass;
    Button mSignUp;

    FirebaseDatabase mDataBase;
    DatabaseReference mRef;
    FirebaseAuth mAuth;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);

        mName = findViewById(R.id.user_name);
        mEmail = findViewById(R.id.user_mail);
        mPass  = findViewById(R.id.user_pass);
        mSignUp = findViewById(R.id.sign_up);

        mDataBase = FirebaseDatabase.getInstance();
        mAuth = FirebaseAuth.getInstance();

        mSignUp.setOnClickListener((v) -> {
                final String name = mName.getText().toString();
                final String email = mEmail.getText().toString();
                String pass = mPass.getText().toString();

                if (name.isEmpty() || email.isEmpty() || pass.isEmpty()){
                    Toast.makeText( context: SignupActivity.this, text: "Please fill all the fields", Toast.LENGTH_SHORT).show();
                }else if (!email.contains("@")){
                    Toast.makeText( context: SignupActivity.this, text: "Please enter a valid email address", Toast.LENGTH_SHORT).show();
                }else {
                    Toast.makeText( context: SignupActivity.this, text: "Signing Up...", Toast.LENGTH_SHORT).show();
```

Figure 9 SignUp Activity

Collect class:

This class reasonable to generate collect number queue and stored it in firebase database.

```
public class CollectActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_collect);
        TextView ticketNum = findViewById(R.id.ticket_num);
        Button done = findViewById(R.id.Done);


        Random random = new Random();
        int n = random.nextInt( bound: 999) + 2000;

        ticketNum.setText(""+n);

        DatabaseReference mRef = FirebaseDatabase.getInstance().getReference( path: "QueueNumberUsers");
        mRef.child(FirebaseAuth.getInstance().getCurrentUser().getEmail().replaceAll( regex: "@" , replacement: "")
        .replaceAll( regex: "\\." , replacement: "")).child("number").setValue(ticketNum.getText().toString());
        done.setOnClickListener((v) → {
                startActivity(new Intent( packageContext: CollectActivity.this , MainActivity.class));
                finish();
        });
    }
}
```

Figure 10 Collect Number Activity

7.1.2 Extensible Markup Language (XML)

XML is generally used to manipulate the layout for each Activity involved in the Queuing Number Application. The coding standards used shall comply with the type of Android layouts applied on each Activity such as Relative Layout, Absolute Layout, or Constraint Layout and also Android specified terms and standards.

Application Code using XML:

Login layout:

This code responsible of the layout and the interface of the login so can be easy for user to use friendly login page and not complicated.

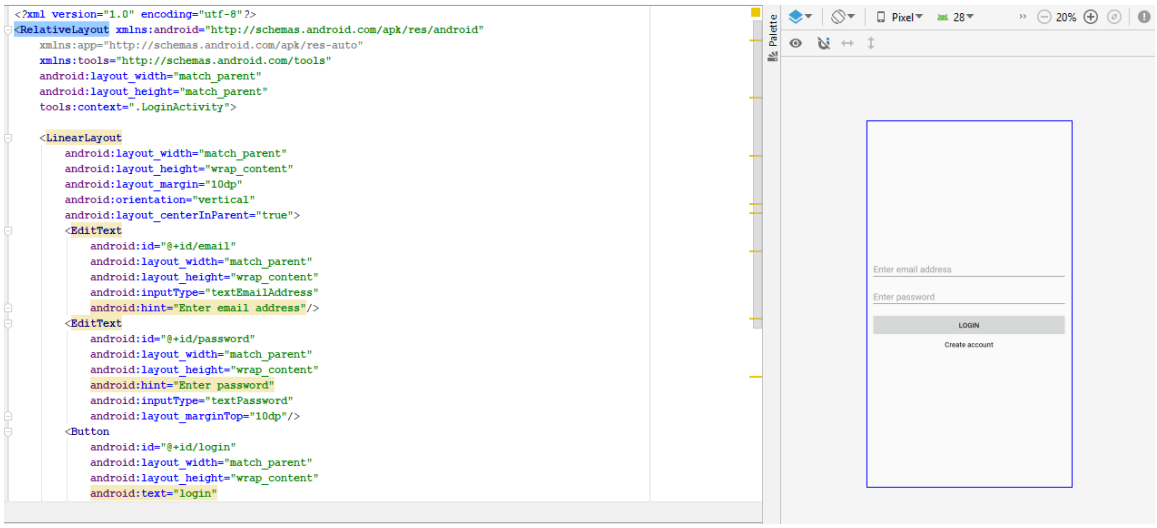


Figure 11 LogIn Layout

Signup layout:

This code is responsible of the layout of the signup using constraint layout so everything appears in the same position to user.

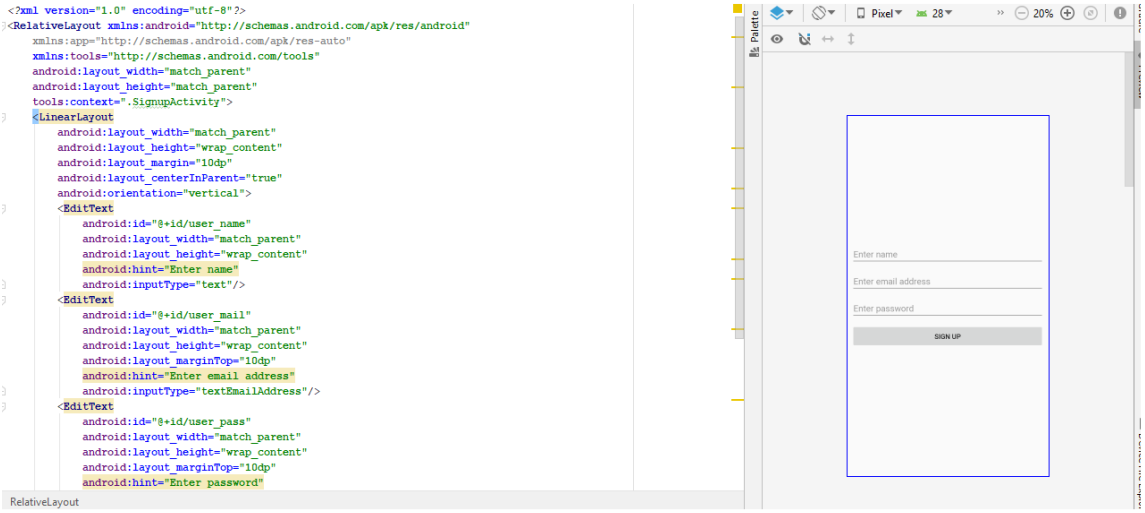


Figure 12 SignUp Layout

7.1.3 Firebase Database

The Firebase Realtime Database is a NoSQL cloud-based database that syncs data across all clients in Realtime, and provides offline functionality. Data is stored in the Realtime database as JSON, and all connected clients share one instance, automatically receiving updates with the newest data.

7.2 *Software Used:*

Software used for the development, implementation and deployment of Queuing Number is Android Studio.

7.2.1 Android Studio:

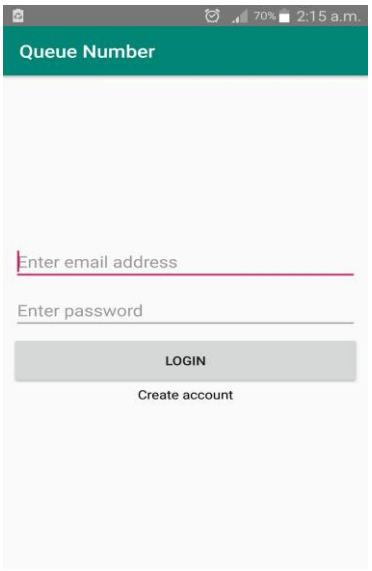
Android Studio is the official Integrated Development Environment (IDE) for Android app development, and includes everything you need to build Android apps. Android Studio offers even more features that enhance your productivity when building Android apps.

7.3 *Screen shots of the application:*

7.3.1 LogIn Functionality

Provides fields to user for input of Email address and Password.

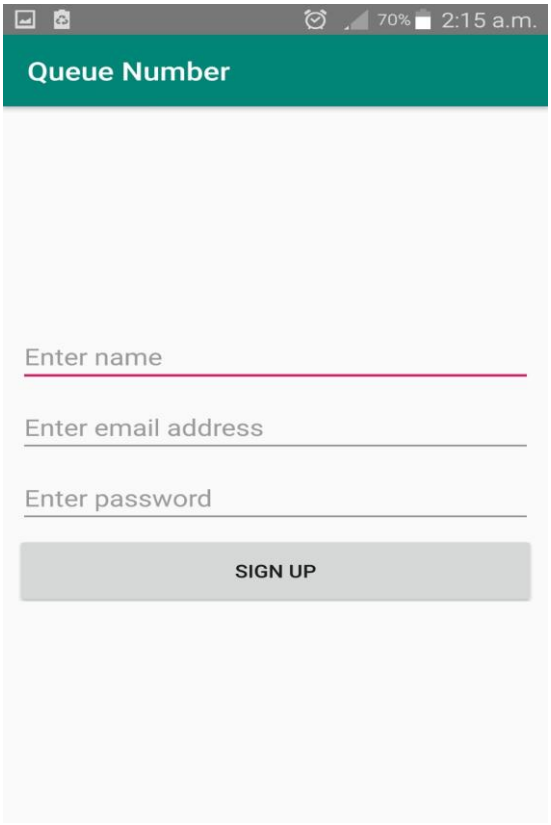
Verification of login credentials will be performed here.



7.3.2 SignUp Functionality

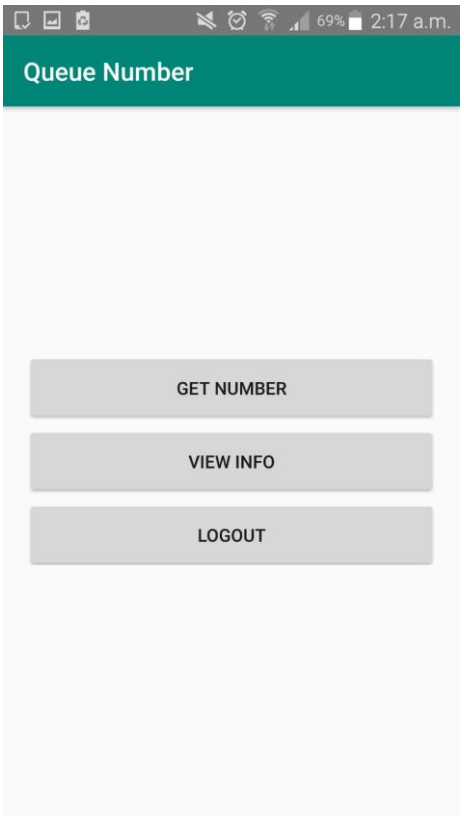
Provides fields to user for input of Name, Email address and Password.

will be performed here, and stored in Firebase Database.



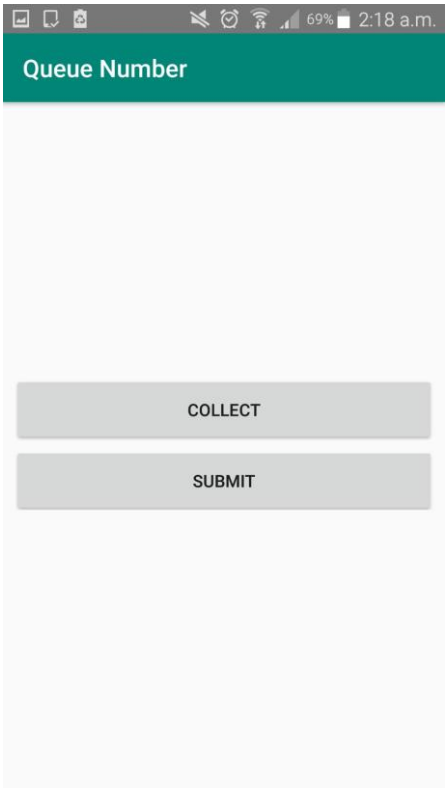
7.3.3 Main page Functionality:

Provides options to user for Get number whether for collect or submit, View Info and Logout.



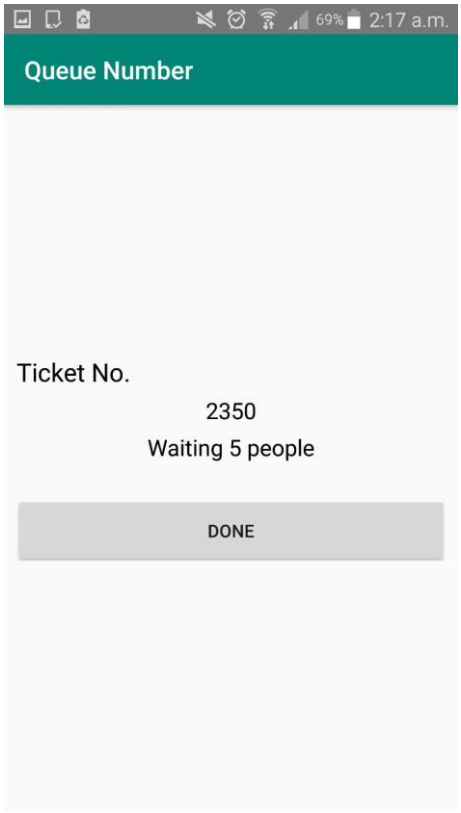
7.3.4 Get Number Functionality:

Provides options to user for collect number or submit number.



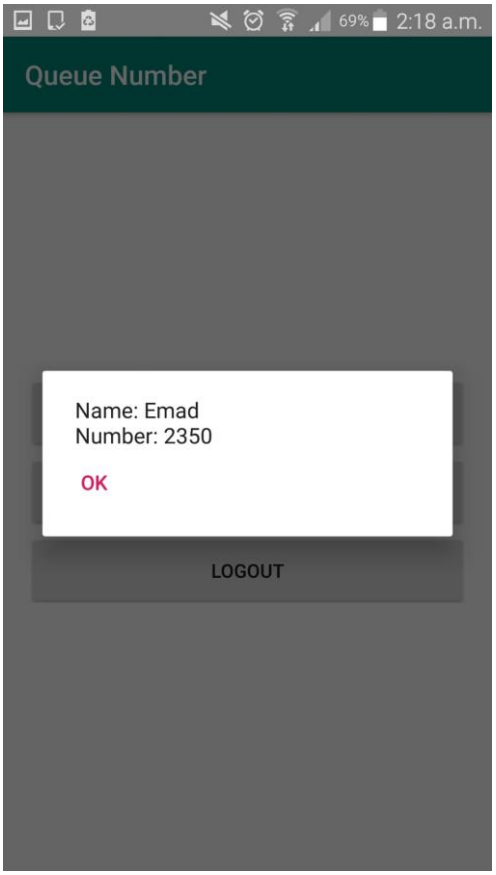
7.3.5 Collect/submit Functionality:

Generate a number whether its for collect or for submit, and also show the waiting people.



7.3.6 View info Functionality:

Provides to user records for Name, and the number of the ticket.



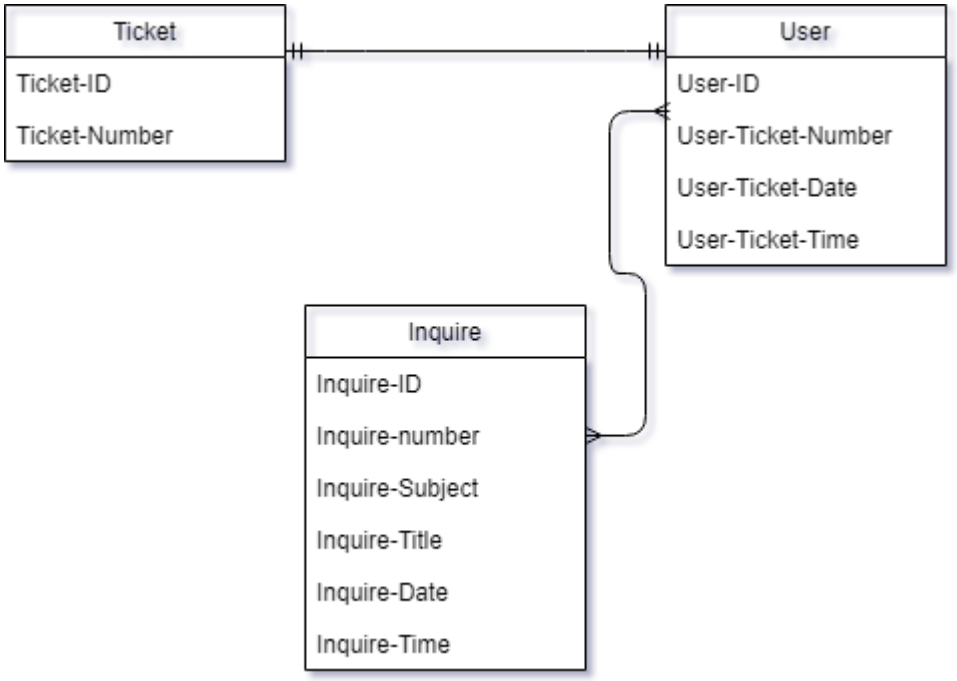
7.4 *Database Structure*

7.4.1 Introduction:

As we all know, behind any fully functional project whether it is an app or even a website there is a database consists of few tables deployed, and for this project the database is created using Firebase Database.

7.4.2 Entity Relationship Diagram (ERD):

This diagram below shows the relationship between sets of entities stored in database, we have three Tables Ticket table, User table and inquire. The relationship between ticket table and user is one to one, and between user table and inquire the relationship is many to many.



7.4.3 Data Dictionary:

Table Name: Ticket

Function: Table that stores Ticket records.

Field Name	Data Type	No Null	Primary Key	Foreign Key	Description
Ticket-ID	Int	Yes	Yes	No	Attribute that stores unique ID number for Ticket.
Ticket-Number	int	Yes	No	Yes	Attribute that uniquely identify the ticket number, and store the ticket number.

Table Name: User

Function: Table that stores User information.

Field Name	Data Type	No Null	Primary Key	Foreign Key	Description
User-ID	int	Yes	Yes	No	Attribute that stores unique User ID.
User-Ticket Number	Int	Yes	No	Yes	Attribute that uniquely identify the User ticket number, and store the User ticket number.
User-Ticket Date	Date	No	No	No	Attribute that uniquely stores the date of User Ticket.
User-Ticket Time	Time	No	No	No	Attribute that uniquely stores the time of User Ticket.

Table Name: Inquire

Function: Table that stores Inquire information.

Field Name	Data Type	No Null	Primary Key	Foreign Key	Description
Inquire-ID	Int	Yes	Yes	No	Attribute that stores unique inquire ID.
Inquire-Title	Varchar (255)	No	No	No	Attribute that uniquely identify and stores inquire title.
Inquire-Date	Date	No	No	No	Attribute that uniquely stores the date of inquire.
Inquire-time	Time	No	No	No	Attribute that uniquely stores the time of inquire.

8 Chapter 8-Conclusion

Conclusion:

- I developed an android application that will allow Students to join a virtual queue via mobile phone.
- allowing students to check there turn without the need to be present at the visa hall.
- providing an estimated time that gives the Students proper idea when their turn will start.
- provide a better overall system to the student.
- reduce wasting time, by show estimate waiting time and sent notification

  when their number is nearing.

9 Reference

1) Ijcaonline.org. (2018). [online] Available at: https://www.ijcaonline.org/archives/volume153/number3/enefiok-2016-ijca-911985.pdf [Accessed 15 Nov. 2018].

2) Ijrter.com. (2018). [online] Available at: https://www.ijrter.com/papers/volume-2/issue-8/queuing-management-using-android-application.pdf [Accessed 15 Nov. 2018].

3) Anon, (2018). [online] Available at: https://www.researchgate.net/publication/313252378_E-Queue_Mobile_Application [Accessed 15 Nov. 2018].

4) Dspace.unimap.edu.my. (2018). [online] Available at: http://dspace.unimap.edu.my/dspace/bitstream/123456789/3057/4/Methodology.pdf [Accessed 15 Nov. 2018].

5) Ijritcc.org. (2018). [online] Available at: http://www.ijritcc.org/download/1424501151.pdf [Accessed 15 Nov. 2018].

6) Shs-conferences.org. (2018). [online] Available at: https://www.shs-conferences.org/articles/shsconf/pdf/2017/01/shsconf_icome2017_00033.pdf [Accessed 15 Nov. 2018].

7) Android Developers. (2018). *Meet Android Studio | Android Developers*. [online] Available at: https://developer.android.com/studio/intro/ [Accessed 15 Nov. 2018].

8) Dart Packages. (2018). *firebase_database | Flutter Package*. [online] Available at: https://pub.dartlang.org/packages/firebase_database [Accessed 15 Nov. 2018].

9) Firebase. (2018). *Firebase Realtime Database | Firebase Realtime Database | Firebase*. [online] Available at: https://firebase.google.com/docs/database/ [Accessed 15 Nov. 2018].

10) En.wikipedia.org. (2018). *Android Studio*. [online] Available at: https://en.wikipedia.org/wiki/Android_Studio [Accessed 15 Nov. 2018].

10 Appendix

Student Survey for Queuing numbering system

Section 1- Android Smart Phone Users Among Limkokwing students

1. Gender

   Male

   Female

2. Faculty

   ----------short answer text----------

3. Semester

   ----------short answer text----------

4. Do you have an android smart phone?

   Yes

   No

5. If yes, which brand do you use?

   Samsung

   LG

   Sony

   Other

6. In a typical weekday, how much time in total do you spend using your mobile?

   Hours

   Minutes

Section 2- Existing services

7. How many times do you go to the visa department?

   Weekly

   Monthly

   Yearly

8. How difficult it is to get a queuing number?

   extremely difficult

   A little difficult

   Not difficult.

9. How much time do you spend until your turn come?

   1 hour

   More than 1 hour

   less than 1 hour

10. How satisfied are you with Visa department services?

    Very satisfied

    Satisfied

    Dissatisfied

Section 3- Study on exceptions

11. What is the most important element to consider in this app, please evaluate the following app attributes.

	Extremely Important	Important	Less important	Not important
Functionality				
Attractiveness				
Friendly user				
Readability				

12. Do you think queuing numbering system will be effective in saving student's time

Yes

Maybe

No

13. please state any recommendations, suggestions or comments that could improve the queuing numbering system application if it were implemented in limkokwing visa department.

----------Long answer text----------
--